

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : 2001-006001

(43) Date of publication of application : 12.01.2001

(51) Int.Cl. G06T 15/70  
G09G 5/00  
G09G 5/36

(21) Application number : 11-172870

(71) Applicant : HITACHI LTD

(22) Date of filing : 18.06.1999

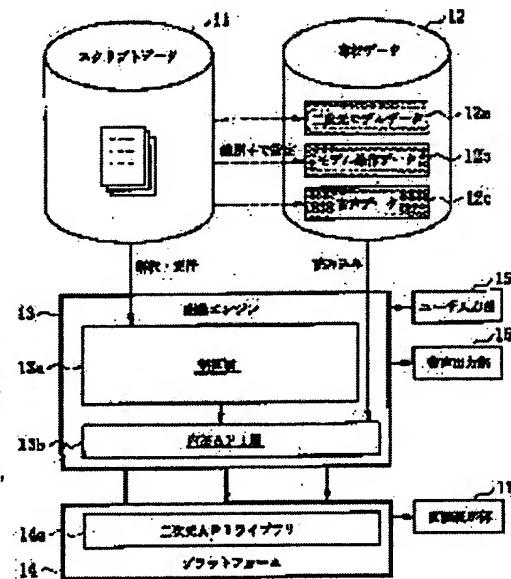
(72) Inventor : MIYAJIMA JUNICHI  
TAKAHASHI AKIO

## (54) THREE-DIMENSIONAL EXPRESSION CONTROL SYSTEM, ITS METHOD AND RECORDING MEDIUM RECORDING ITS PROCESSING PROGRAM

### (57) Abstract:

**PROBLEM TO BE SOLVED:** To provide three-dimensional expression control system/method, which make the operation speed of walk through in a 3D animation game to be constant in spite of the data quantity of a background and which displays data as in a way similar to the scenario that the describer of a script intends.

**SOLUTION:** Even in a computer different in ability (plotting processing operation speed of a screen is slow), time used for the display of a prescribed operation is previously set in a scenario for displaying a picture following the scenario in same time and a control layer 13a controls the update of the screen so that the operation is completed within setting time. When screen update time exceeds setting time, time which is set next by the scenario is reduced by excess time, the execution time of the scenario is reduced and delay from real time is absorbed.



### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

**BEST AVAILABLE COPY**

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-6001

(P2001-6001A)

(43)公開日 平成13年1月12日(2001.1.12)

(51)Int.Cl.  
G 0 6 T 15/70  
G 0 9 G 5/00  
5/36

識別記号  
5 5 0  
5 1 0

F I  
C 0 6 F 15/62  
C 0 9 G 5/00  
5/36

テ-コ-ト\*(参考)  
3 4 0 K 5 B 0 5 0  
5 5 0 B 5 C 0 8 2  
5 1 0 V

審査請求 未請求 請求項の数7 OL (全 15 頁)

(21)出願番号 特願平11-172870

(22)出願日 平成11年6月18日(1999.6.18)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 宮崎 淳一

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

(72)発明者 高橋 明男

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

(74)代理人 10007/274

弁理士 磯村 雅俊 (外1名)

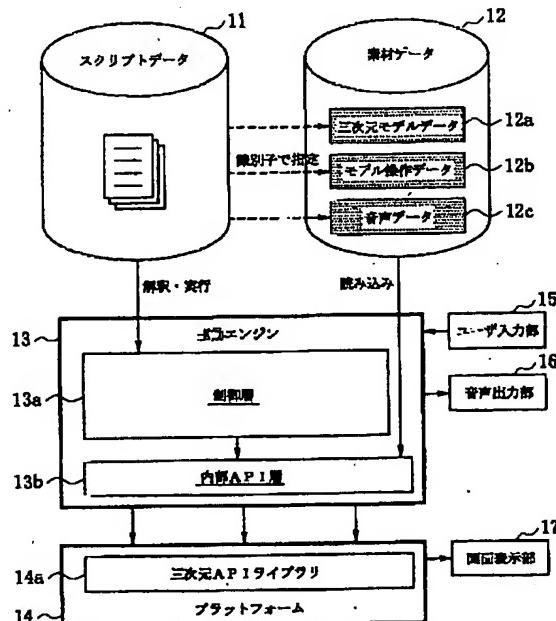
最終頁に続く

(54)【発明の名称】三次元表現制御システムと方法およびその処理プログラムを記録した記録媒体

(57)【要約】

【課題】従来、コンピュータの性能や描画対象のデータ量の増減により、3Dアニメーションの表示速度が変動してしまう。

【解決手段】性能の異なる(画面の描画処理動作速度の遅い)計算機であっても同じ時間でシナリオに従った画像を表示させるために、シナリオにおいて予め、ある動作の表示に用いる時間を設定しておき、制御層13aにより、この設定時間内で動作が完了するように、画面の更新を制御する。もし、画面更新時間が設定時間を超えた場合には、超過時間だけ、シナリオで次に設定されている時間を削減して、シナリオでの実行時間を減らして実時間からの遅れを吸収する。



【特許請求の範囲】

【請求項1】 スクリプト言語で記述されたシナリオに従って三次元モデルのアニメーション表示をコンピュータを用いて行う三次元表現制御システムであって、上記シナリオで予め設定されている時間定義情報と動作定義情報を読み込み、該時間定義情報で定められた時間（定義時間）で上記動作定義情報で定められた動作を完了させるよう画面の更新を制御する制御手段を有することを特徴とする三次元表現制御システム。

【請求項2】 請求項1に記載の三次元表現制御システムにおいて、上記制御手段は、上記画面の更新に要した時間と上記定義時間とを比較して、上記画面の更新時間が上記定義時間を超過した場合、上記シナリオにおいて次の時間定義情報で定められている定義時間から上記超過した時間だけ減らした時間を求め、該求めた時間で、次の動作定義情報で定められた動作を完了させるよう画面の更新を制御することを特徴とする三次元表現制御システム。

【請求項3】 請求項2に記載の三次元表現制御システムにおいて、上記制御手段は、上記超過時間が上記次に定められている定義時間をさらに超過した場合、上記次に定められている時間定義情報での処理を放棄して、上記シナリオにおいてさらに次の時間定義情報で定められている定義時間から上記さらに超過した時間だけ減らした時間を求め、該求めた時間で、上記シナリオにおいてさらに次の動作定義情報で定められた動作を完了させるよう画面の更新を制御することを特徴とする三次元表現制御システム。

【請求項4】 スクリプト言語で記述されたシナリオに従って三次元モデルのアニメーション表示をコンピュータを用いて行う三次元表現制御方法であって、上記シナリオで予め設定されている時間定義情報と動作定義情報を読み込むステップと、該ステップで読み込んだ上記時間定義情報で定められた時間（定義時間）で上記動作定義情報で定められた動作を完了させるよう画面の更新を制御するステップとを有することを特徴とする三次元表現制御方法。

【請求項5】 請求項4に記載の三次元表現制御方法において、上記画面の更新に要した時間と上記定義時間とを比較するステップと、該ステップでの比較の結果、上記画面の更新時間が上記定義時間を超過していれば、上記シナリオにおいて次の時間定義情報で定められている定義時間から上記超過した時間だけ減らした時間を求めるステップとを有し、該求めた時間で、次の動作定義情報で定められた動作を完了させるよう画面の更新を制御することを特徴とする三次元表現制御方法。

【請求項6】 請求項5に記載の三次元表現制御方法において、上記超過時間と上記次に定められている定義時間とを比較するステップと、該ステップでの比較の結果、上記超過時間が上記次に定められている定義時間を

さらに超過していれば、上記次に定められている時間定義情報での処理を放棄し、上記シナリオにおいてさらに次の時間定義情報で定められている定義時間から上記さらに超過した時間だけ減らした時間を求めるステップとを有し、該求めた時間で、上記シナリオにおいてさらに次の動作定義情報で定められた動作を完了させるよう画面の更新を制御することを特徴とする三次元表現制御方法。

【請求項7】 スクリプト言語で記述されたシナリオに従って三次元モデルのアニメーション表示をコンピュータを用いて行うシステムによる三次元表現制御方法の処理手順プログラムを記録する記録媒体であって、請求項4から請求項6のいずれかに記載の三次元表現制御方法における各ステップを、上記コンピュータに実行させるためのプログラムを記録したことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、コンピュータによる三次元画像の表示技術に係り、特に、スクリプト記述を用いた三次元画像のアニメーション（3Dアニメーション）表示の制御を効率的に行うのに好適な三次元表現制御システムと方法およびその処理プログラムを記録した記録媒体に関するものである。

【0002】

【従来の技術】 従来、三次元画像は、静止画として組み立てて構成するだけでも、専用のアプリケーションや、専門の技術者によるデータ作成が必要であった。さらに、三次元画像を利用したアニメーション（3Dアニメーション）の表現や、仮想空間内を自由に歩く（ウォータースルー）などのインターラクティブな動作を行うソフトウェアアプリケーションの開発は、三次元の演算などの専門知識が要求される。この結果、三次元画像の利用は専門技術者の多い家庭用ゲーム機や、CAD（Computer Aided Design）などの特定の分野に限られていた。

【0003】 このような3Dアニメーションの表現や、インターラクティブなソフトウェアアプリケーションを開発する場合、ハードウェアに密着した低レベルな機能を提供する三次元表示ライブラリを用いてC言語やアセンブラー言語などでプログラミングする。この三次元表示ライブラリを利用するには、専門的な三次元技術が必要とされる。

【0004】 また、3Dアニメーション等において、一般的のユーザが目にして興味を引くような表現は、映画などの映像業界で培われる技術である。そのため、表現力のある三次元アプリケーションを制作するためには、三次元表示ライブラリを利用するための専門的な三次元技術と、映画などの映像業界で培われる技術の両方を持った技術者が必要となる。

【0005】 このような三次元画像を、専門家でない一

般のユーザーにも手軽に利用できるようにするために、スクリプトなどの簡易言語で三次元画像を作成する技術がいくつか提案されている。例えば、国際標準規格(ISO/IEC14772)のVRML(Virtual Reality Modeling Language)は、簡易言語で三次元モデルを記述し、ブラウザなどの専用アプリケーション上で、その言語内容を解釈して表示するものである。また、特開平9-330425号公報や特開平9-34675号公報等においても、スクリプトを利用した三次元画像の表示技術が開示されている。

【0006】このスクリプトを利用した三次元画像の表示技術により、いわゆるパーソナルコンピュータを用いた一般のユーザーによる3Dアニメーションの作成等も可能である。尚、VRMLを用いてパーソナルコンピュータで三次元画像を表示する技術に関しては、例えば、「インターフェース [特集]インターネットとマルチメディア技術 1996年3月号」(1996年、CQ出版社発行)の第134~146頁に記載されている。

【0007】しかし、パーソナルコンピュータには、搭載しているCPU(Central Processing Unit)やグラフィックボード等の性能差により、画像表示処理能力の高いものや画像表示処理能力の低いものがある。例えば3Dアニメーションのアプリケーションプログラムであれば、画像表示処理能力の低いパーソナルコンピュータでは高速なアニメーション表示がなされず、プログラムの作成者が意図した動作は表現できない。

【0008】また、画像表示処理能力の高いパーソナルコンピュータであっても、各画面(フレーム)を構成するデータ量の増減に伴い、動作が不安定になる。例えば、いわゆるウォームスルーレーの動作において、データ量の少ない背景から、データ量の多い背景に切り換わった場合、データ量の多い背景での各フレーム1枚毎の描画に時間を要してしまい、その動作が遅くなってしまう。

【0009】このようにウォームスルーレーの画面上での動作が、背景の切り換わりで遅くなると、例えば、ゲーム中であれば、操作しているユーザーは、その動作がスローダウンすることにより、何か新たな変化の発生(新たな敵の出現等)を予見してしまい、ゲーム製作者の考えているシナリオと異なるものとなってしまう。

【0010】また、画像表示処理能力の低いパーソナルコンピュータにおいて、ある程度高速なアニメーション表示をさせるには、装置性能に合わせてフレームを間引きする従来技術があるが、この従来技術では、均一にフレームが間引かれるだけであり、例えば上述のウォームスルーレーの画面上での動作におけるデータ量の少ない背景からデータ量の多い背景に切り換わった時点での動作の変動を完全に吸収することはできない。

【0011】

【発明が解決しようとする課題】解決しようとする問題点は、従来の技術では、コンピュータの性能や描画対象

のデータ量の増減により、3Dアニメーションの表示速度が変動してしまう点である。

【0012】本発明の目的は、これら従来技術の課題を解決し、例えば3Dアニメーションゲームにおけるウォームスルーレー等の動作速度を背景のデータ量の大小にかかわらず一定とし、スクリプトの記述者が意図したシナリオ通りに表示することを可能とする三次元表現制御システムと方法およびその処理プログラムを記録した記録媒体を提供することである。

【0013】

【課題を解決するための手段】上記目的を達成するため、本発明の三次元表現制御システムと方法は、性能の異なる(画面の描画処理動作速度の遅い)計算機であっても同じ時間でシナリオに従った画像を表示させるために、シナリオにおいて予め、ある動作の表示に用いる時間を設定しておき、この設定時間内で動作が完了するよう、画面の更新を制御する。

【0014】もし、画面更新時間が設定時間を超えた場合には、超過時間だけ、シナリオで次に設定されている処理時間を削減することにより、シナリオでの実行時間を減らし、また、超過時間が次の設定時間も超過していれば(T)、この設定時間での処理を放棄して、さらに次の設定時間での処理を、時間(T)分だけ削減した時間で行い、実時間からの遅れを吸収する。

【0015】尚、3Dモデルデータ、モーションデータ、サウンドデータ、ビットマップデータ等の全てのコンテンツを形成する素材データと、各素材データの振る舞いやコンテンツの流れを表現したシナリオを記述したスクリプトデータとを相互に分けて記憶し、各素材データにはそれぞれ固有の識別子を付与し、この識別子によりスクリプトデータで各素材データを特定する構成として、三次元画像の利用に専門的な技術を不要とする。

【0016】また、スクリプトの解釈処理を行う処理部と、実際の表示を行う処理部とを分け、装置のハードウェア構成やオペレーティングシステムの種別などのプラットフォームに依存する部分を、実際の表示を行う処理部に吸収して移植性を高める。

【0017】

【発明の実施の形態】以下、本発明の実施の形態を、図面により詳細に説明する。図1は、本発明の三次元表現制御システムの本発明に係る構成の一実施例を示すブロック図であり、図14は、そのコンピュータ装置構成例を示すブロック図である。

【0018】図14において、141はCRT(Cathode Ray Tube)やLCD(Liquid Crystal Display)あるいはプロジェクタ等からなる表示装置、142はキーボードやマウス等からなる入力装置、143はハードディスクドライブ等からなる外部記憶装置、144はCPU(Central Processing Unit)や主メモリ等を有し蓄積プログラム方式によるコンピュータ処理を行なう情報処

理装置、145は本発明に係る処理プログラムやデータ等を記録する光ディスク、146は光ディスク145の読み取り動作を行なう駆動装置である。

【0019】光ディスク145に記録された処理プログラムやデータをインストールして主メモリに読み込むことにより、情報処理装置144内に、図1に示す三次元表現制御システムが構築される。以下、図1を用いて三次元表現制御システムの詳細を説明する。

【0020】図1に示すように、本例の三次元表現制御システムは、アプリケーションの内容を記述したスクリプトデータ11と、実際に表示に使用される素材データ12、それらを読み込み解釈実行を行う駆動エンジン13、駆動エンジン13での解釈実行結果に基づき三次元画像データを生成する表示処理部（図中、「プラットフォーム」と記載）14、図14における入力装置142からの指示を入力するユーザ入力部15、スピーカーなどの音響機器を介して音声を出力する音声出力部16、表示処理部14で生成された三次元画像データに基づき図14における表示装置141を介して三次元画像を表示出力する画面表示部17により構成されている。

【0021】三次元画像の表示は、技術的には数学知識を必要とする反面、表現としては人間の実生活に一番近く、アプリケーションを表現する場合に必要な要素は、「人物モデルが登場する」、「演技する」、「話す」などのように絞り込みが可能であり、残りの基本的な表示技術は、パッケージ化が可能があるので、本例では、素材データ12として、基本的な表示技術をパッケージ化している。

【0022】この素材データ12は、アプリケーションで利用される三次元モデルデータ12a、モデル操作データ12b、音声データ12cで構成され、それぞれには識別子が対応付けられている。これにより、三次元(3D)モデルデータやモデル操作(モーション)データ、サウンド(音声)データ等、コンテンツを形成する素材データを、各素材データの振る舞いやコンテンツの流れを表現したシナリオを記述したスクリプトデータ11と分けて記憶し、スクリプトデータ11では、識別子により各素材データを特定することができる。

【0023】このように、三次元で表現する内容と三次元モデルデータが独立しているので、三次元画像の利用に専門的な技術が不要となる。また、従来の三次元表示ライブラリを用いた技術のように、アプリケーションの内容が固定的になり、きめ細やかな表現が出来なくなるばかりでなく、三次元画像を外部出力の一つとして他のシステムから利用することも不便となるといった問題がなくなり、三次元モデルの再利用や事前に相手先にある三次元モデルを利用することなどが可能となる。さらに、同じアプリケーションを他のプラットフォームに移植する際も、表現内容に関わる部分がプログラムで記述されておらず、大幅な変更作業は不要である。

【0024】また、本例の三次元表現制御システムでは、簡易言語としてスクリプトを用いてアプリケーションの内容を記述し、そのスクリプトを解釈実行する駆動エンジン13を利用する。この駆動エンジン13は、スクリプトデータ11の解釈実行を行う制御層13aと、データのハンドリングと三次元APIライブラリ14aへのアクセスを行い実際の表示を行う内部API層13bの二階層構成とし、プラットフォームに依存する部分は内部API層13bに吸収し、駆動エンジン13としての移植性を高めている。

【0025】駆動エンジン13の制御層13aは、スクリプトデータ11を解釈実行して、素材データ12を読み込み処理を行う際、識別子を用いて指定する。また、制御層13aには、三次元アプリケーションで汎用的に利用可能な機能として、後に図7～図11で詳細に説明するアニメーションの実時間管理機能と仮想三次元空間内を自由に動くこと(ウォータースルー)などができる仮想空間内の移動制御機能を用意する。

【0026】移動制御機能は、アニメーション時の三次元モデルの移動や、ユーザ入力部15からのユーザ入力による仮想空間内の移動機能(ウォータースルー)を実現するものであり、実時間管理機能は、マシンスペックによらずに、指定した時間でアニメーションを再生したり、ユーザ入力による仮想空間内の一定の速度での移動を保つものであり、それぞれ本発明に係る三次元表現制御を行うものである。

【0027】以下、図2を用いて駆動エンジン13の内部構成を説明する。図2は、図1における駆動エンジンの構成例を示すブロック図である。本例の駆動エンジン(13)の制御層13aには、スクリプトデータ11の解釈実行を行うスクリプト解釈実行モジュール21と、移動制御モジュール22、ライト制御モジュール23、カメラ制御モジュール24、三次元モデル制御モジュール25、サウンド制御モジュール26が設けられている。

【0028】この制御層13aにおいて、サウンド制御モジュール26以外は、スクリプトから動的に必要な数だけ生成される。起動時はスクリプト解釈実行モジュール21のみが動作し、スクリプト内で必要な各種モジュールを動的に生成・リンクし、アプリケーションにあつた実行形態を取る。

【0029】また、内部API層13bには、ユーザ入力API27、三次元作画API28、サウンドAPI29が設けられており、ユーザ入力API27は、インターラクティブなアプリケーションなどの場合、移動制御モジュール22に、図1のユーザ入力部15からのユーザ入力情報を伝達する。三次元作画API28は、ライト制御モジュール23、カメラ制御モジュール24、三次元モデル制御モジュール25からの情報を元に三次元画面を作画し画面を更新する。そして、サウンドAPI

29は、サウンド制御モジュール26から再生すべき音声データ12cの識別子を指定する形で制御され、音声出力部16を通じて音声を再生する。

【0030】次に、図3を用いて図1におけるスクリプトデータ11と素材データ12との対応付けについて説明する。図3は、図1におけるスクリプトデータと素材データとの対応付けに用いられるキーワード・テーブルの構成例を示す説明図である。スクリプトは、オブジェクト指向的に記述され、三次元仮想空間内の振る舞いを表現する。

【0031】本図3のキーワード テーブル30において、機能モジュールを生成するキーワード31では、仮想空間内のオブジェクトを指定するキーワードが登録されており、機能モジュール内で使用されるパラメータ設定キーワード32では、各オブジェクトの制御内容を設定するキーワードが登録されており、スクリプト全体で使用されるキーワード33では、その他のキーワードが登録されている。

【0032】図1の駆動エンジン13は、図2のスクリプト解釈実行モジュール21により、これらのキーワードを解釈実行して、対応する制御モジュールを生成する。キーワード テーブル30における主なキーワードは、対応した各種モジュールで処理されるが、スクリプト全体で使用されるキーワード33のWaitキーワード33aは、後の図7で説明する実時間管理機能で処理される。このようなキーワードを用いて次の図4～図6に示すようなスクリプトを作成する。

【0033】以下、図4～図6により、キーワードを用いたスクリプトの作成例と、そのスクリプトに基づく図1の駆動エンジン13による表示制御動作例を説明する。

【0034】図4は、本発明に係るスクリプト記述例とそれに基づく表示制御例の第1例を示す説明図である。本図4に示す例のスクリプト記述例41の(a)の内容に基づき図2のスクリプト解釈実行モジュール21は、三次元モデル制御モジュール25を生成し、三次元モデル制御モジュール25は、読み込む三次元モデルデータ12aを特定し、その表示位置と方向を順に設定する。

【0035】また、スクリプト記述例41の(b)の内容に基づき図2のスクリプト解釈実行モジュール21は、カメラ制御モジュール24を生成し、カメラ制御モジュール24は、三次元仮想空間でのユーザの視点としてのカメラの位置と方向を設定する。そして、スクリプト記述例41の(c)の内容に基づき図2のスクリプト解釈実行モジュール21は、ライト制御モジュール23を生成し、ライト制御モジュール23は、現実世界と同じように物体が見えるようにするため、光(ライト)の位置と方向を設定する。これにより、スクリプト表示例42で示すように各制御対象が配置され、その結果として、スクリプト表示例43に示すような一体の三次元モデルが

表示される。

【0036】図5は、本発明に係るスクリプト記述例とそれに基づく表示制御例の第2例を示す説明図である。本例は、仮想空間内を自由に移動出来るアプリケーションをスクリプトで記述した例を示す。本例のスクリプト記述例51の(a)で示す部分で、画面に表示する環境モデル(「環境モデルA」)を設定する。

【0037】また、スクリプト記述例51の(b)で示す部分で、仮想空間内のユーザの視点を表現するモジュールを生成する。そして、生成したモジュールに移動開始地点の初期位置(「Position」)と初期方向(「Rotation」)を設定する。その結果、「環境モデルA」で表現される三次元仮想空間の中を、キーボードやジョイスティックなどの外部入力機器によるユーザ入力によって移動する(ウォークスルー)アプリケーションが駆動される。すなわち、スクリプト表示例52～54の順に、ユーザ入力により前進した場合の画面の変化を表示する。

【0038】図6は、本発明に係るスクリプト記述例とそれに基づく表示制御例の第3例を示す説明図である。本例は、3Dアニメーションのアプリケーションをスクリプトでシナリオ形式で実際に記述した例を示す。アニメーションは、時系列に「Waitキーワード33a」を挟みながら、登場人物や、その台詞、カメラの動きなどを記述する。スクリプト記述例61における(a)では、登場人物の歩く演技と移動(5秒間)を設定している。そして、「Wait<2秒>」のキーワードにより、2秒間の待ちを設定し、(b)では、「音声1」の再生を設定している。これにより、画面上での登場人物Aの5秒間の歩行途中、その歩行開始の2秒後に「音声1」が再生される。

【0039】さらに、次の「Wait<1秒>」のキーワードにより、1秒間の待ちが設定され、その後、(c)で、挨拶の演技(「演技2」と「音声2(よろしくお願ひします)」の再生が設定され、その後「Wait<2秒>」のキーワードにより、2秒間の待ちが設定されている。この一連の処理結果として、スクリプト表示例62～64に順に示されるように、登場人物Aがこちらに5秒間、「位置1」まで歩きながら挨拶をするアニメーションが再生される。

【0040】次に、このようなスクリプトに基づく図1の駆動エンジン13(図2のスクリプト解釈実行モジュール21)による本発明に係る三次元画像の表示制御に関して、図7～図11を用いて説明する。図7は、図2のスクリプト解釈実行モジュールの本発明に係る実時間管理を行う機能構成例を示すブロック図である。実時間管理は、実時間コマンド処理モジュール72と実時間補間機能73で実現される。

【0041】実時間コマンド処理モジュール72では、後述の図10で説明するように、画面更新処理モジュール74からの情報(「前回の画面更新からの実時間の差

分」に基づき、本発明に係る実時間に対応したコマンド(スクリプトからの各種制御モジュールへのコマンド71)の「読み飛ばしと読み込み休止」を行う。

【0042】まず、実時間補間機能73の説明を図8を用いて行う。図8は、図7における実時間補間機能による動作例を示す説明図である。図8(a)に示すように、三次元空間内の点Aから点Bに移動するコマンドを処理する場合において、スクリプト上での移動開始時間を図8(b)におけるT11、また、その移動終了時間をT13とする。

【0043】この時、画面更新が正常にT11、T12、T13の割り込み(画面に再描画が可能となるタイミング)で行われた場合、T12での移動位置を、補間された座標として計算し、画面表示に反映する。すなわち、点Aの座標ベクトルをva、点Bの座標ベクトルをvb、T12での座標ベクトルをv12とすると、次の数1の式で計算される。

【数1】

$$v_{12} = v_a + \frac{T_{12} - T_{11}}{T_{13} - T_{11}} (v_b - v_a)$$

【0044】これが基本的な補間となる。しかし、アニメーション再生中は、再生装置の描画処理速度に影響されて、スクリプトの指定する開始時間と終了時間が、画面更新時間とずれことが多い。例えば、図8(b)のT21、T22、T23、T24の間に画面更新の割り込みがされることがある。

【0045】この場合、T21の時点では移動コマンドが読み込まれていないため、何も行われない。T22の時点で、移動座標を補間する。ここでは、点Aの座標ベクトルをva、点Bの座標ベクトルをvb、T22での座標ベクトルをv22とすると、次の数2の式でT22での座標ベクトルv22が計算される。

【数2】

$$v_2 = v_1 + r v_e$$

この結果がカメラ制御モジュール24に送られ、表示に反映される。

【0051】次に、図10を用いて、図1における三次元表現制御システムの本発明に係る処理動作を説明する。図10は、図1における三次元表現制御システムによる本発明の三次元表現制御方法に係る処理動作例を示すフローチャートである。

【0052】本例は、図1の三次元表現制御システムにおいて、スクリプトデータ11を駆動エンジン13を用いて解釈実行する制御プロセスを示しており、図7で説明した実時間管理に係る実行コマンド処理モジュール72による「実時間でのコマンドの読み飛ばしと読み込み休止」処理はこのプロセスで実現される。尚、実際のコ

$$v_{22} = v_a + \frac{T_{22} - T_{11}}{T_{13} - T_{11}} (v_b - v_a)$$

【0046】同様に、T23での座標ベクトルをv23とすると、次の数3の式でT23での座標ベクトルv23が計算される。

【数3】

$$v_{23} = v_a + \frac{T_{23} - T_{11}}{T_{13} - T_{11}} (v_b - v_a)$$

【0047】そして、T24の時点では、移動終了时刻は過ぎているため、点Bの座標を設定する。これらの処理により、スクリプト上の記述に沿った実時間上でのあるべき位置に座標を設定することができる。回転についても同様な補間を行う。

【0048】次に、図1の駆動エンジン13(図2のスクリプト解釈実行モジュール21)の本発明に係る三次元画像の表示制御における実時間管理を行う他の機能構成例を、図9を用いて説明する。

【0049】図9は、図2のスクリプト解釈実行モジュールの本発明に係る実時間管理を行う他の機能構成例を示すブロック図である。本例は、駆動エンジン13での移動制御モジュール(22)による仮想空間内を自由に移動する場合の実時間補間を行うものであり、実時間補間機能73は、ユーザ入力からの移動制御モジュール(22)へのコマンド(「回転速度の設定」、「移動速度の設定」)と、画面更新処理モジュール74からの情報(「前回の画面更新からの差分の時間」)に基づき、カメラ制御モジュール24を制御する。

【0050】例えば、現在の座標ベクトルをv1、移動後の座標ベクトルをv2、現在向いている方向の単位ベクトルをve、速度値をrとすると、移動後の座標ベクトルv2は、次の数4の式で計算される。

【数4】

コンピュータでの時計機能で計測する画面更新処理時間に要する実経過時間を「T」、「Wait」キーワードで設定される待ち時間を「t」、シナリオでの経過時間と実経過時間との差分である仮想経過時間を「t」とする。

【0053】駆動エンジン13は、まず、仮想経過時間t=0と設定する初期化処理を行う(ステップ101)。次に、前回の画面更新開始時からの現在の実時間までの差分である実経過時間Tを設定する(ステップ102)。最初はこの実経過時間Tも「0」が設定される。そして、図1のスクリプトデータ11の解釈を開始し(ステップ103)、まずスクリプトの一行を読み込み(ステップ104)、その行の頭にあるキーワードを判定する(ステップ105)。

【0054】キーワードが、モジュール生成キーワードであれば(ステップ106)、モジュール識別子の取得とモジュール生成を行い(ステップ117)、また、キーワードが、パラメータ設定キーワードであれば(ステップ107)、パラメータの取得とモジュールへの設定を行い(ステップ118)、キーワードが、Soundキーワードであれば(ステップ108)、サウンドデータ識別子の取得と再生処理を行い(ステップ119)、その後、ステップ104に戻り、次の行の読み取りを行う。

【0055】また、キーワードが、Waitキーワードであれば(ステップ109)、そのWaitキーワードで設定されている待ち時間t'の値を取得し(ステップ110)、「 $t = t + t'$ 」として、新たな仮想経過時間tを算出する(ステップ111)。最初は、ステップ101での初期化処理において「 $t = 0$ 」が設定されており、例えば、図6に示したスクリプト例では、「Wait <2秒>」となっており、新たな仮想経過時間tは「 $0 + 2$ 」で2秒となる。

【0056】このようにして算出した新たな仮想経過時間tと、ステップ102で設定された実経過時間Tとの比較を行う(ステップ112)。ここでは、新たな仮想経過時間tは2秒、ステップ102で設定された実経過時間Tは「0」で、「 $t > T$ 」であり、実時間に対する遅れはないものと判断し、仮想経過時間tから実経過時間Tを引いた時間をさらに新たな仮想経過時間tに設定した後(ステップ113)、ステップ117～119で取得した各モジュールの実経過時間Tにおける更新処理(ステップ114)と、それらに基づく画面更新処理(ステップ115)を行う。この実経過時間Tにおける更新処理と画面更新処理は、図8で説明した図7の実時間補間機能73の処理に基づき行われる。

【0057】この各モジュールの更新処理と画面更新処理に要した時間を計測して、新たな実経過時間Tとして設定する(ステップ116)。そしてステップ112に戻り、この新たな実経過時間Tと、ステップ113で設定したさらなる新たな仮想経過時間tとの比較を行い、実経過時間Tが仮想経過時間tより大きくなるまで、すなわち、実時間に対する遅れが発生するまで、それぞれの実経過時間Tでの各モジュールの更新処理と画面更新処理を繰り返す。

【0058】そして、ステップ112において、実経過時間Tが仮想経過時間tより大きくなり、実時間に対する遅れが発生すると、その差分( $T - t$ )を算出して、その差分を新たな実経過時間Tとして設定し(ステップ120)、ステップ104の処理に戻る。このようにして、算出した差分( $T - t$ )を新たな実経過時間Tとして設定することにより、シナリオにおいて次に設定されている動作に対するステップ112～116での処理時間(画面更新に用いる時間)が短縮され、実時間に対する

遅れの分を吸収することができる。

【0059】このようなステップ112～116、120の処理により、画面更新処理に要する時間に応じて、画面の更新回数を増減すること、すなわち、シナリオの実行時間を制御することができ、例えば、背景のデータ量が多くなり描画速度が遅くなった際にも、次のスクリプト行への処理に移行する時間の遅れを最小に抑えることができ、画面を構成するデータ量の増減による動作の不安定さ(変動)を吸収することができる。

【0060】さらに、ステップ114の更新処理とステップ115の画面更新処理に大幅な時間を要した結果、ステップ120で設定した新たな実経過時間Tが、次のスクリプトに対する処理におけるステップ111での仮想経過時間tよりも大きくなった場合には、そのスクリプト部分に対してのステップ114、115での各モジュールの更新処理と画面更新処理を行うことなく、次のスクリプト部分の処理に移行する。すなわち、「実時間でのコマンドの読み込み休止」を、実時間に追いつくまで行う。このようにして、シナリオで設定されている時間と実時間とを一致させて、設定された動作を完了させることができる。

【0061】次に、図11を用いて、図10のステップ114における各モジュールの更新処理の動作を説明する。図11は、図1における三次元表現制御システムの本発明に係る処理動作例を示すフローチャートである。本例は、特に図7における実時間補間機能73による実時間管理に係る処理動作例を示している。

【0062】まず、図3に示したキーワード・テーブル30におけるモジュールを生成するキーワード31(「User」、「Model」、「Camera」、「Light」、….)によって、モジュールの生成を行う(ステップ201)。次に、図7における画面更新処理モジュール74からの情報(前回の画面更新からの実時間の差分)に基づき、実経過時間Tの設定を行い(ステップ202)、その後、モジュール更新処理を開始する(ステップ203)。

【0063】ここでは、図7の実時間コマンド処理モジュール72からの補間移動のコマンド入力に基づき(ステップ213)、新規補間処理の有無を判定する(ステップ204)。有る場合は、さらに実行中の補間処理の有無を判断し(ステップ205)、有る場合は、実行中の補間終了値の設定を行ってから(ステップ206)、コマンドの遅れ時間差による補間値の修正(ステップ207)と、残り補間時間tの設定(ステップ208)、および、実経過時間Tと残り補間時間tとの比較(ステップ209)を行う。

【0064】また、ステップ204の新規補間処理の有無の判別処理において、新規補間処理が無い場合は、さらに、実行中の補間処理の有無を判断し(ステップ214)、有る場合に、ステップ209における実経過時間

$T$ と残り補間時間 $t$ との比較を行う。無い場合は、補間処理は無いということになり、ステップ202に戻り、実経過時間 $T$ の設定を行う。

【0065】ステップ209での実経過時間 $T$ と残り補間時間 $t$ との比較において、残り補間時間 $t$ が大きい場合は、補間中と判断し、現在の補間値の算出(ステップ210)と、残り補間時間の更新(ステップ211)を行う。また、ステップ209で実経過時間 $T$ が大きい場合は、補間は既に終了していると判断し、補間終了値の設定(ステップ215)と、補間処理終了(ステップ216)を行う。そして、結果としての値を三次元作画AP1に設定して(ステップ212)、画面を更新する。

【0066】次に、図1の三次元表現制御システムにおけるスクリプトの記述を容易とする例を図12と図13を用いて説明する。図12は、本発明の三次元表現制御システムの他の構成例を示すブロック図であり、図13は、図12における三次元表現制御システムで表示される画面表示例を示す説明図である。

【0067】図12に示す例は、図1における三次元表現制御システムに、スクリプトデータ 素材データ作成編集インタフェース120を設けたものであり、このスクリプトデータ 素材データ作成編集インタフェース120は、図13に示すスクリプトデータ 素材データ作成編集インタフェース画面130をディスプレイ上に表示する。

【0068】この図13に示すスクリプトデータ 素材データ作成編集インタフェース画面130は、スクリプトデータ格納フォルダ131、素材データ格納フォルダ132、スクリプトテンプレート133、素材テンプレート134、スクリプト編集操作ウィンドウ135、スクリプトパラメータ入力サブウィンドウ136からなり、フィル イン ザ ブランク(Fill in the blank)方式(可変部分を穴明き状態にして各穴を埋めることによりスクリプトデータを完成させる方式)での入力が可能となっており、ユーザは、このスクリプトデータ 素材データ作成編集インタフェース画面130により、スクリプトデータ11を作成、編集したり、素材データ12の各データを追加、削除することができる。

【0069】以上、図1～図14を用いて説明したように、本例の三次元表現制御システムと方法では、スクリプトで記載されているシナリオに従って三次元画像を画面に表示する場合、シナリオで予め設定されている時刻での経過と、コンピュータで計数している実時間での時刻の経過とを、例えば画像の更新毎に比較し、シナリオでの経過時刻が実時間より遅れていれば、次の画像の更新をキャンセル等して次の命令の実行に移ることにより、その遅れを補正する。

【0070】このように、本例では、実時刻と、現在表示している画面が表示されるべきシナリオの時刻とを比較し、実時間から遅れていれば、シナリオの実行時間を

減らすので、性能の異なる(画面の描画処理動作速度の遅い)計算機であっても、同じ時間でシナリオに従った画像を表示することができ、また、データ量の増減にかかわらずシナリオ通りの表示時間制御ができる。

【0071】また、本例では、スクリプトのような簡易言語によって内容を記述することで、三次元画像表示の専門知識を必要最小限とし、アプリケーション作成の飛躍的な生産性向上が可能となる。また、簡易言語により、アプリケーションをプラットフォームから独立させることで、変更無しで幅広いプラットフォーム上で再現可能なアプリケーションを作成可能となる。さらに、三次元モデルをアプリケーション内容の情報から独立させることで、伝達すべき情報が少なくなり、ネットワーク上で三次元を利用したアプリケーションの配信が可能となる。

【0072】また、本例では、スクリプトデータ 素材データ作成編集インタフェース画面130により、スクリプト一覧や素材データ一覧からドラッグ&ドロップでスクリプトが作成できるので、短時間でスクリプトデータを作成でき、テンプレートを容易したことによりさらに短時間で、スクリプトを作成できる上に誤りを大幅に削減できる。

【0073】尚、本発明は、図1～図14を用いて説明した実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。例えば、本例では、本発明の三次元表現制御システムと方法を実装する装置として特に特定していないが、いわゆるパーソナルコンピュータやワークステーションを用いることができる。また、本発明の三次元表現制御方法に係る処理プログラム(ソースコードあるいは実行コード)やデータを記録する媒体としては、FD(Flexible Disk)やDVD(Digital Video Disc)を用いることでも良い。さらに、本発明の三次元表現制御方法に係る処理プログラムやデータを、通信回線を介してパーソナルコンピュータやワークステーション等に配信してインストールすることでも良い。

【0074】

【発明の効果】本発明によれば、性能の異なるコンピュータであっても同じ時間でシナリオに従った画像表示ができる、また、描画対象のフレームのデータ量の増減による三次元画像の表示速度の変動を吸収すること等ができる、例えば3Dアニメーションゲームにおけるウォークスルー操作中に、データ量の少ない背景からデータ量の多い背景に切り換わった時点でも、ウォークスルー動作は変化せず、スクリプトの記述者が意図したシナリオ通りに表示させることができ、三次元表現技術を用いる装置の性能の向上を図ることが可能である。

【図面の簡単な説明】

【図1】本発明の三次元表現制御システムの本発明に係る構成の一実施例を示すブロック図である。

【図2】図1における駆動エンジンの構成例を示すブロック図である。

【図3】図1におけるスクリプトデータと素材データとの対応付けに用いられるキーワード・テーブルの構成例を示す説明図である。

【図4】本発明に係るスクリプト記述例とそれに基づく表示制御例の第1例を示す説明図である。

【図5】本発明に係るスクリプト記述例とそれに基づく表示制御例の第2例を示す説明図である。

【図6】本発明に係るスクリプト記述例とそれに基づく表示制御例の第3例を示す説明図である。

【図7】図2のスクリプト解釈実行モジュールの本発明に係る実時間管理を行う機能構成例を示すブロック図である。

【図8】図7における実時間補間機能による動作例を示す説明図である。

【図9】図2のスクリプト解釈実行モジュールの本発明に係る実時間管理を行う他の機能構成例を示すブロック図である。

【図10】図1における三次元表現制御システムによる本発明の三次元表現制御方法に係る処理動作例を示すフローチャートである。

【図11】図1における三次元表現制御システムの本発明に係る処理動作例を示すフローチャートである。

【図12】本発明の三次元表現制御システムの他の構成例を示すブロック図である。

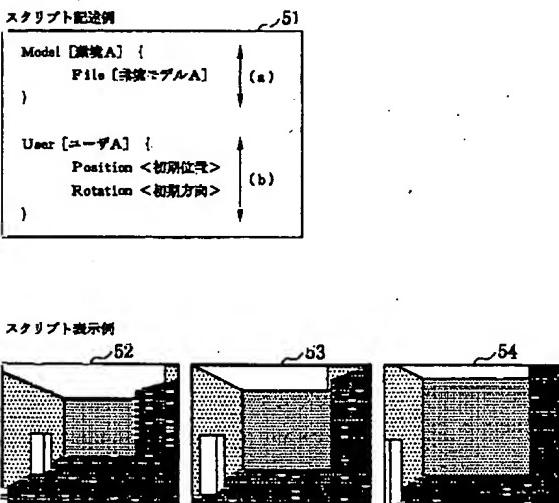
【図13】図12における三次元表現制御システムで表示される画面表示例を示す説明図である。

【図14】図1における三次元表現制御システムを実装したコンピュータ装置構成例を示すブロック図である。

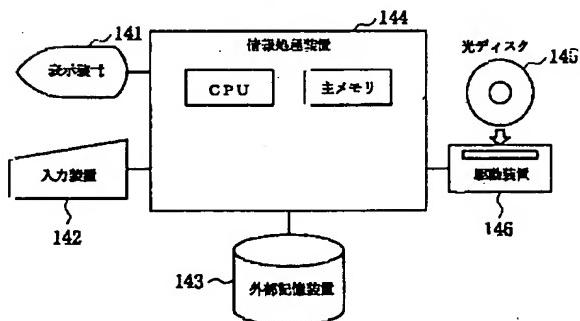
【符号の説明】

11:スクリプトデータ、12:素材データ、12a:三次元モデルデータ、12b:モデル操作データ、12c:音声データ、13:駆動エンジン、13a:制御層、13b:内部API層、14:表示処理部(プラットフォーム)、14a:三次元APIライブラリ、15:ユーザ入力部、16:音声出力部、17:画面表示部、21:スクリプト解釈実行モジュール、22:移動制御モジュール、23:ライト制御モジュール、24:カメラ制御モジュール、25:三次元モデル制御モジュール、26:サウンド制御モジュール、27:ユーザ入力API、28:三次元作画API、29:サウンドAPI、30:キーワードテーブル、31:機能モジュールを生成するキーワード、32:機能モジュール内で使用されるパラメータ設定キーワード、33:スクリプト全体で使用されるキーワード、41:スクリプト記述例、42,43:スクリプト表示例、51:スクリプト記述例、52~54:スクリプト表示例、61:スクリプト記述例、62~64:スクリプト表示例、71:スクリプトからの各種制御モジュールへのコマンド、72:実時間コマンド処理モジュール、73:実時間補間機能、74:画面更新処理モジュール、91:ユーザ入力からの移動制御モジュールへのコマンド、120:スクリプトデータ・素材データ作成編集インタフェース、130:スクリプトデータ・素材データ作成編集インタフェース画面、131:スクリプトデータ格納フォルダ、132:素材データ格納フォルダ、133:スクリプテンプレート、134:素材テンプレート、135:スクリプト編集操作ウィンドウ、136:スクリプトパラメータ入力サブウィンドウ、141:表示装置、142:入力装置、143:外部記憶装置、144:光ディスク、145:駆動装置、146:光ディスクドライブ。

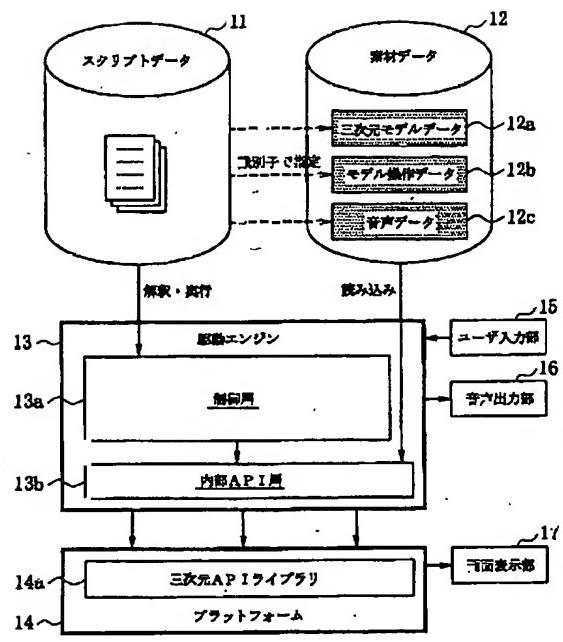
【図5】



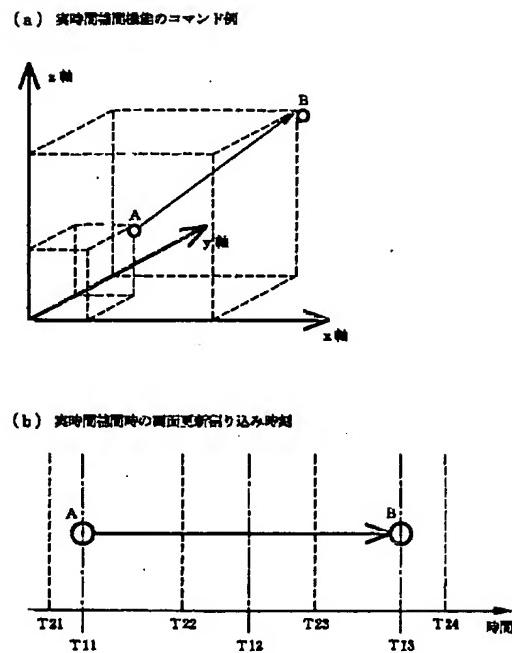
【図14】



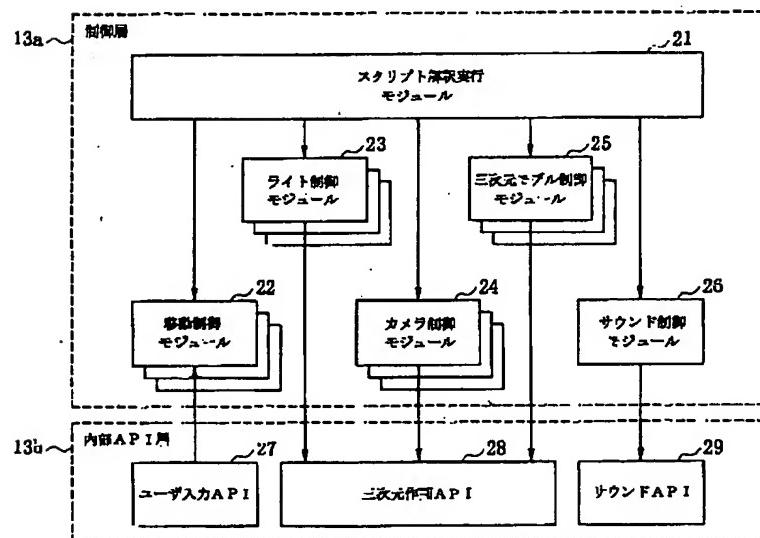
【図1】



【図8】



【図2】



【図3】

30 キーワード・ナーブル

初期モジュールを生成するキーワード		
制御対象	キーワード	使用する素材データ
ユーザーの位置	User	
三次元モデル	Model	三次元モデルデータ モデル操作データ
カメラ	Camera	
ライト	Light	

31

初期モジュール内で使用されるパラメータ設定キーワード		
設定項目	キーワード	パラメータ
仮想空間内の位置	Position	仮想空間内の位置
仮想空間内の方向	Rotation	仮想空間内の方向
位置移動	Move	目的の位置と移動時間
方向回転	Turn	目的の方向と回転時間
モデルの操作	Perform	素材データの識別子
素材データ指定	File	素材データの識別子
モジュールの連結	Attach	対象モジュール

32

スクリプト全体で使用されるキーワード		
制御対象	キーワード	使用する素材データ
音声再生	Sound	音声データ
待機時間の設定	Wait	

33

33a ～待機時間の設定

【図4】

スクリプト記述例 41

```

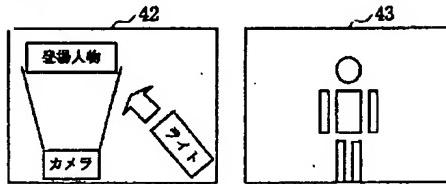
Model【登場人物A】 {
  File【人物モデル】
  Position <位置1>
  Rotation <方向1>
}

Camera【カメラA】 {
  Position <位置2>
  Rotation <方向2>
}

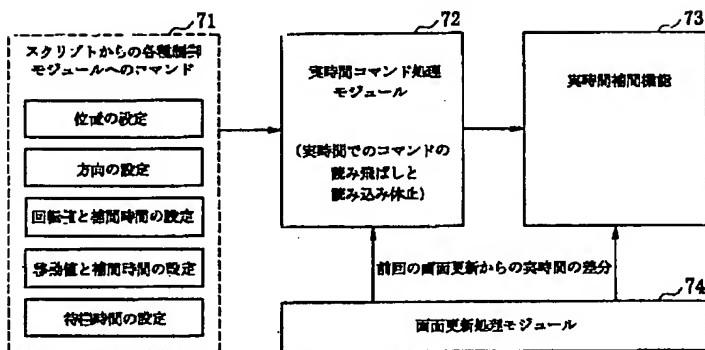
Light【ライトA】 {
  Position <位置3>
  Rotation <方向3>
}

```

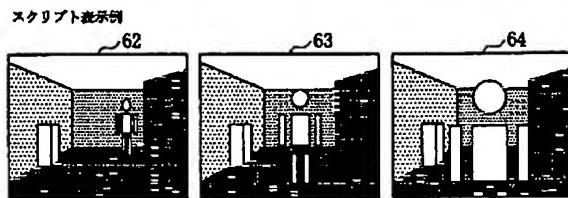
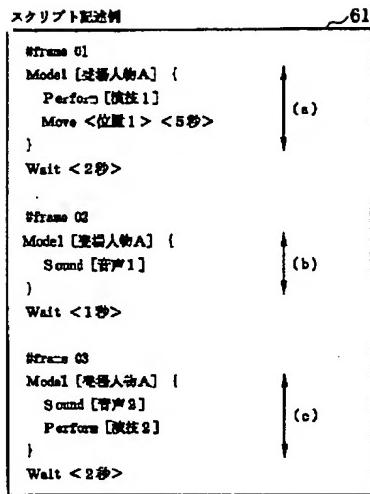
スクリプト表示例



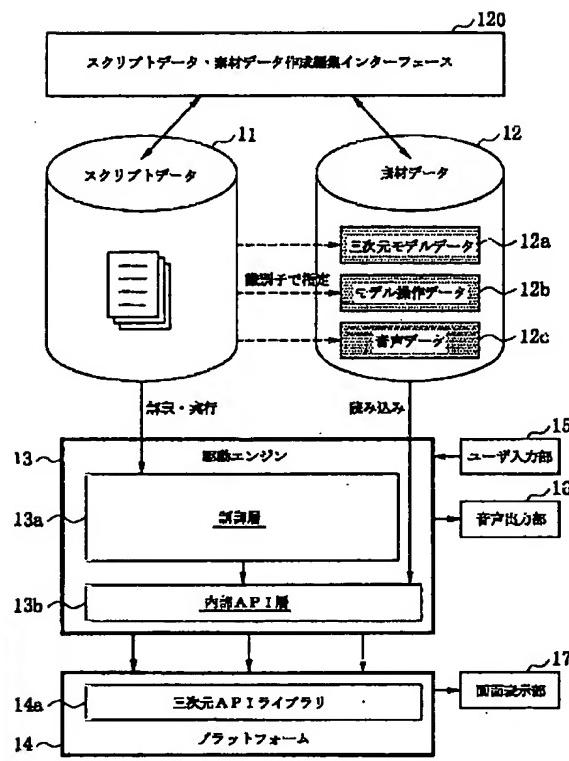
【図7】



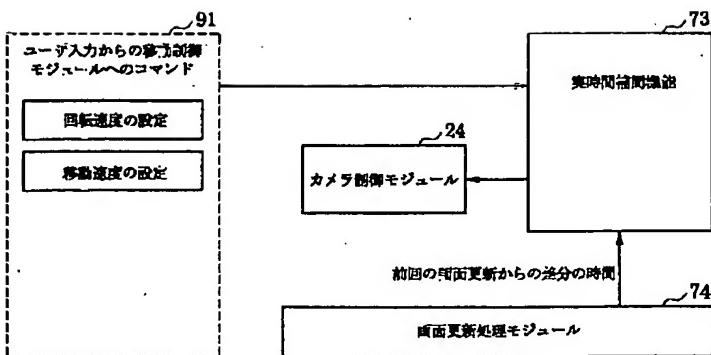
【図6】



【図12】

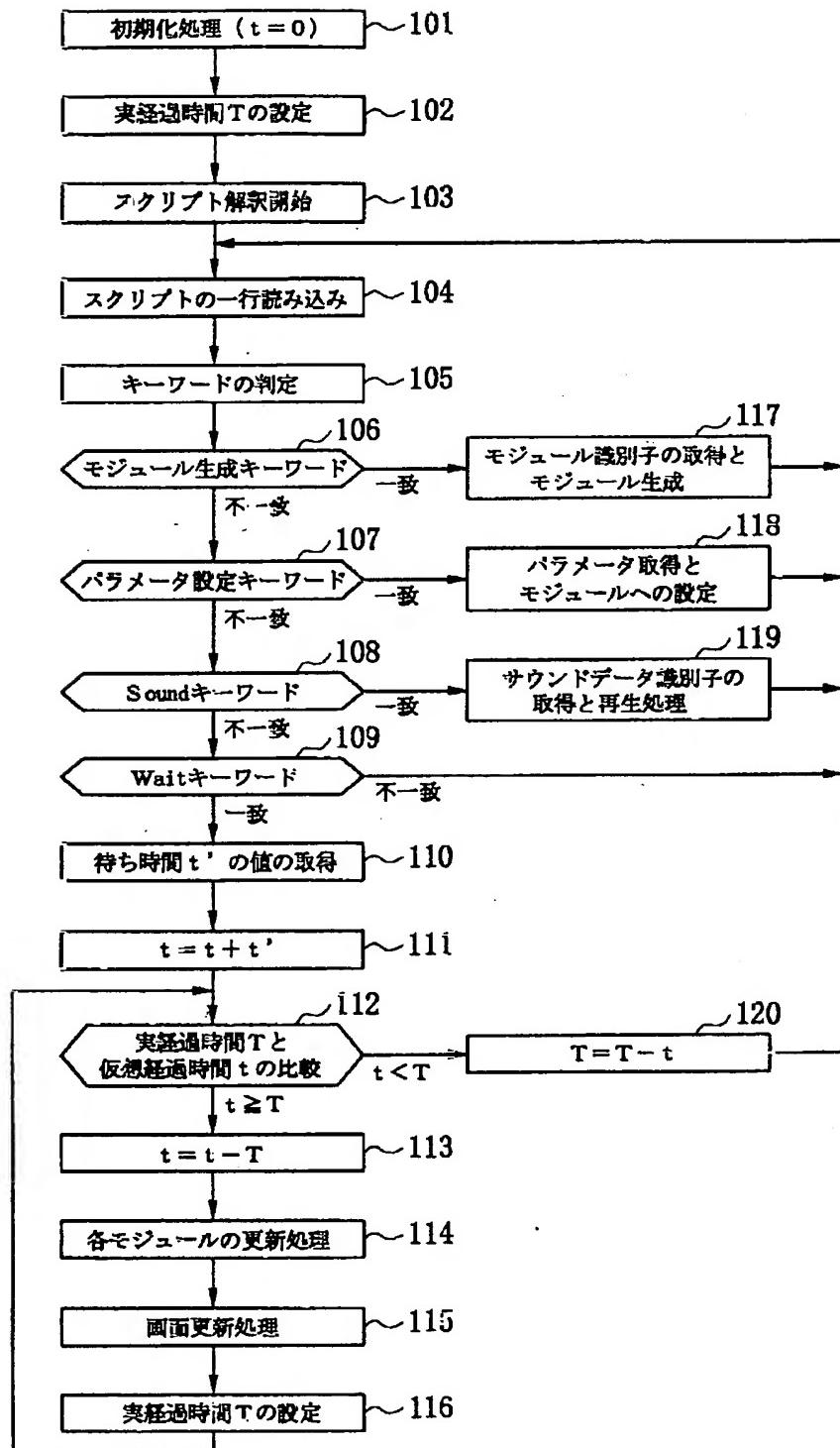


【図9】

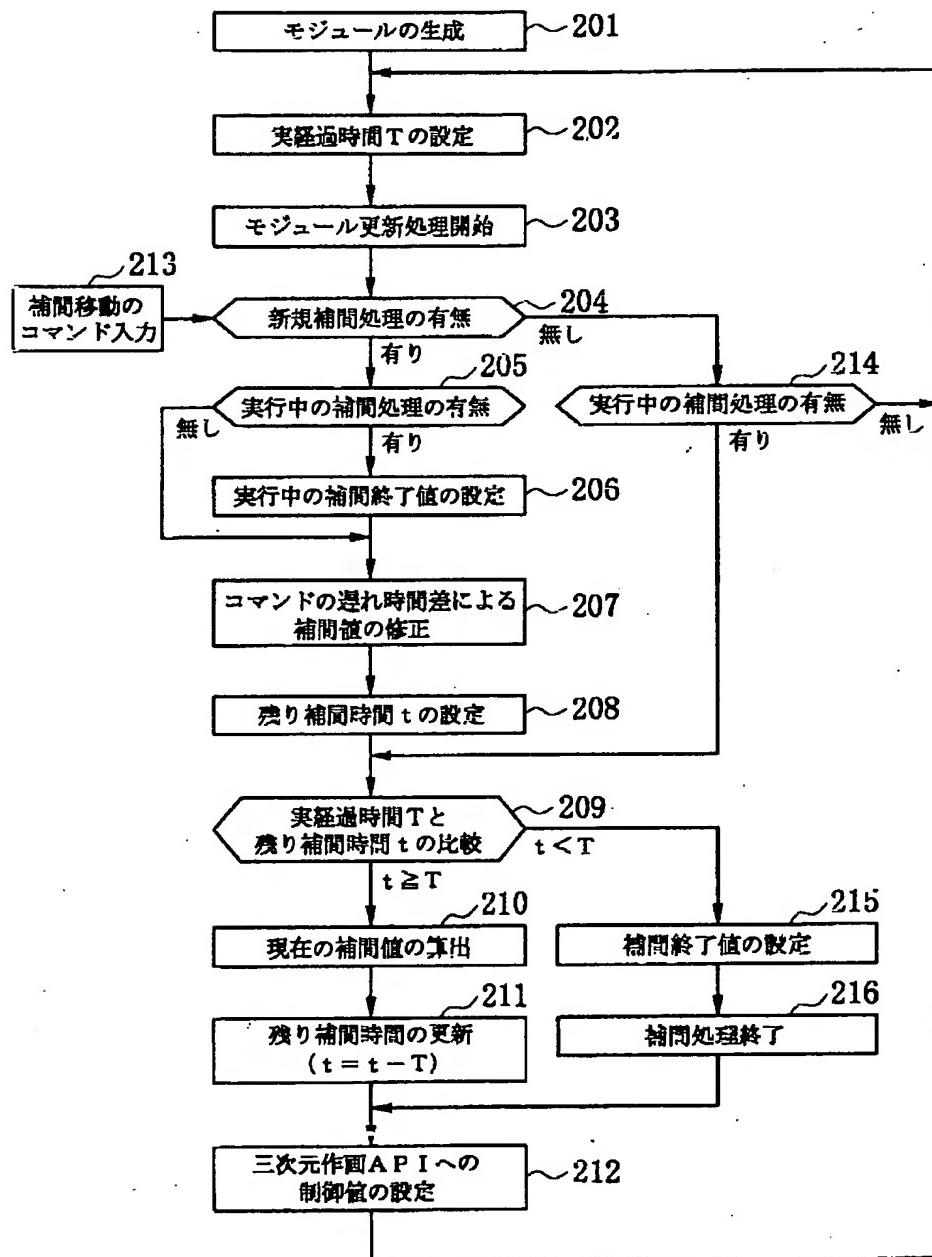


BEST AVAILABLE COPY

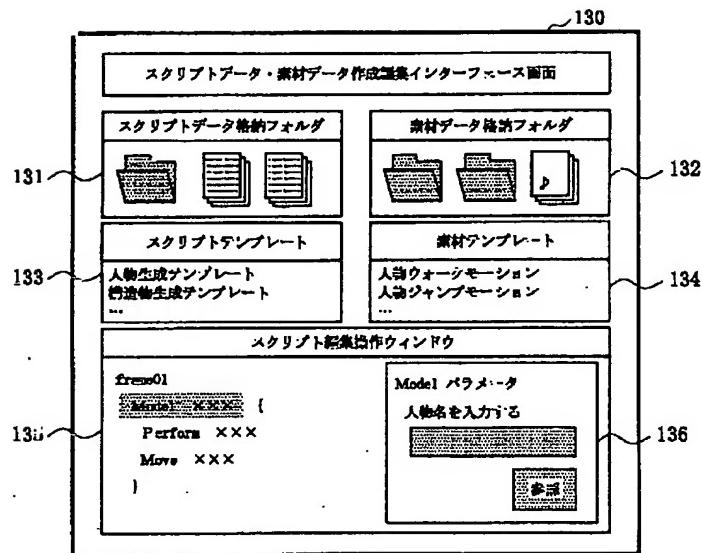
【図10】



【図11】



【図13】



---

フロントページの続き

Fターム(参考) 5B050 BA08 BA09 DA10 EA24 EA28  
FA02 FA10  
5C082 AA01 BA43 BA46 CB01 DA87  
MM06 MM10

BEST AVAILABLE COPY